

Requested Patent: JP2001022709A

Title:

CLUSTER SYSTEM AND COMPUTER-READABLE STORAGE MEDIUM STORING
PROGRAM ;

Abstracted Patent: JP2001022709 ;

Publication Date: 2001-01-26 ;

Inventor(s): MURATA AKIFUMI; KOISHI MAKOTO ;

Applicant(s): TOSHIBA CORP ;

Application Number: JP19990198971 19990713 ;

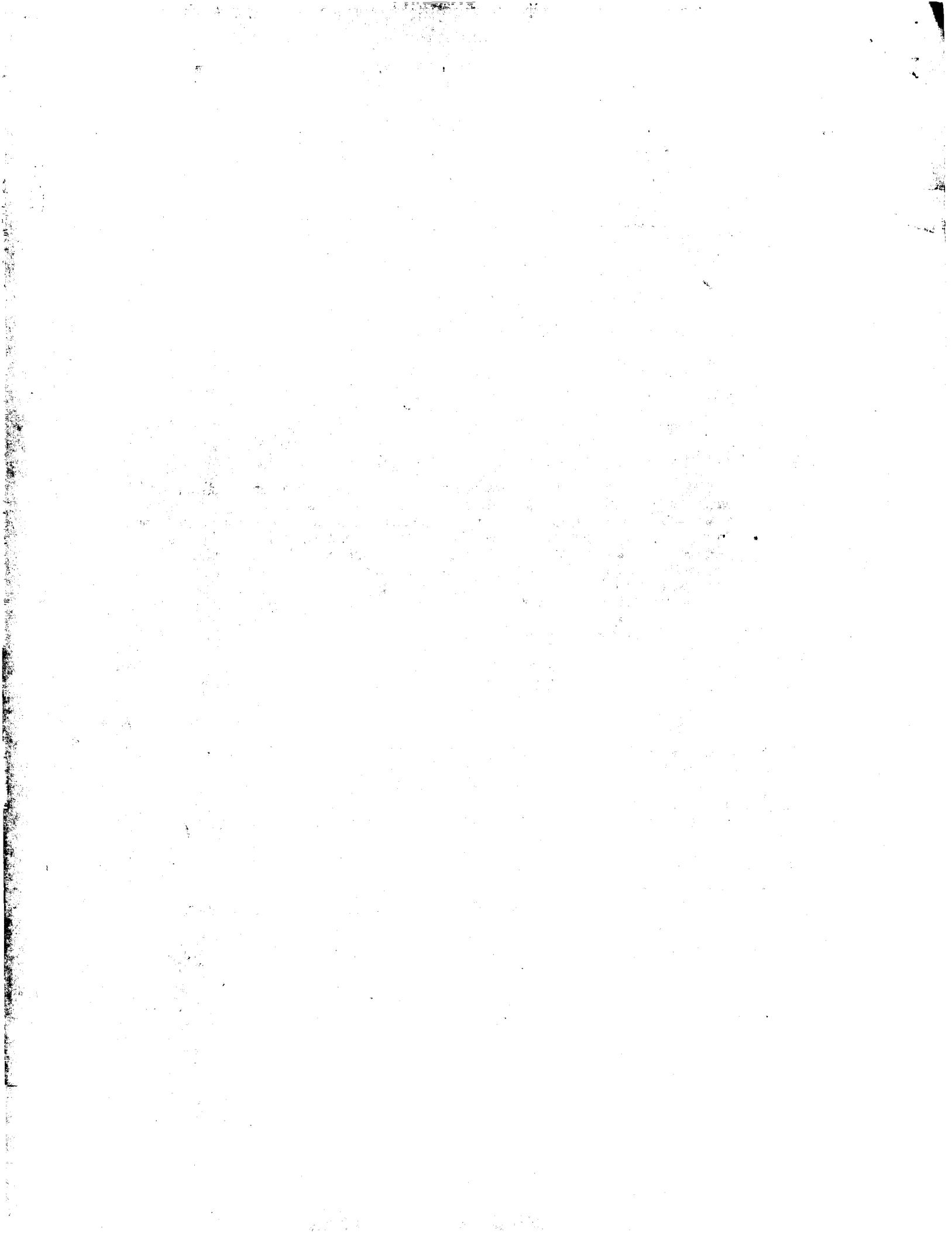
Priority Number(s): ;

IPC Classification: G06F15/16; G06F9/46; G06F11/20; G06F11/30; G06F15/177 ;

Equivalents: ;

ABSTRACT:

PROBLEM TO BE SOLVED: To easily introduce a new program into a cluster system and to continuously execute the program even if some abnormality occurs.
SOLUTION: In the cluster system which monitors the operation states of programs 6a to 6c running on computers 2a and 2b, an identification information acquiring means 11a acquires identification information of the programs 6a to 6c and a monitor means 12a monitors whether or not the programs 6a to 6c indicated by the acquired identification information are normal. When it is judged that a monitored program is abnormal, a restarting means 12b restarts the abnormal program on the computer 2a where the abnormal program was executed. Once the abnormality of the restarted program is detected, a program transfer means 11b executes the programs 6a to 6c on the computer 2a where the abnormal program was executed, on the other computer 2b.



(19)日本国特許庁 (JP)

(12) 公開特許公報 (A)

(11)特許出願公開番号

特開2001-22709

(P2001-22709A)

(43)公開日 平成13年1月26日(2001.1.26)

(51)Int.Cl. ⁷	識別記号	F I	テ-マコト [*] (参考)
G 0 6 F 15/16 9/46	6 4 0 3 3 0 3 6 0	G 0 6 F 15/16 9/46	6 4 0 A 5 B 0 3 4 3 3 0 C 5 B 0 4 2 3 6 0 B 5 B 0 4 5
11/20	3 1 0	11/20	3 1 0 F 5 B 0 9 8
11/30	3 0 5	11/30	3 0 5 J

審査請求 未請求 請求項の数4 OL (全11頁) 最終頁に続く

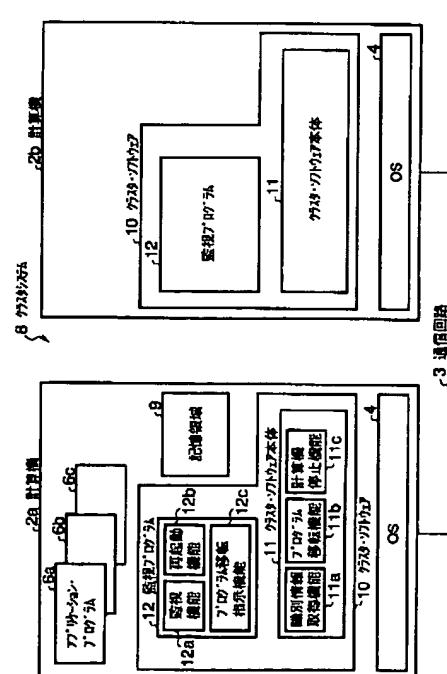
(21)出願番号	特願平11-198971	(71)出願人	000003078 株式会社東芝 神奈川県川崎市幸区堀川町72番地
(22)出願日	平成11年7月13日(1999.7.13)	(72)発明者	村田 明文 東京都府中市東芝町1番地 株式会社東芝 府中工場内
		(72)発明者	小石 誠 東京都府中市東芝町1番地 株式会社東芝 府中工場内
		(74)代理人	100058479 弁理士 鈴江 武彦 (外6名)
			最終頁に続く

(54)【発明の名称】 クラスタシステム及びプログラムを記憶したコンピュータ読み取り可能な記憶媒体

(57)【要約】

【課題】クラスタシステムに容易に新規のプログラムを導入可能とし、何らかの異常が発生しても継続してプログラムを実行可能とする。

【解決手段】複数の計算機2a、2b上で動作するプログラム6a～6cの動作状況を監視するクラスタシステム8であり、プログラム6a～6cの識別情報を識別情報取得手段11aによって取得し、取得された識別情報の示すプログラム6a～6cが正常か否かを監視手段12aが監視する。この監視によってプログラムが異常とされると、再起動手段12bが異常プログラムを実行していた計算機2a上でこの異常プログラムを再起動させる。再起動されたプログラムの異常が検出されると、プログラム移転手段11bが当該異常プログラムを実行していた計算機2a上のプログラム6a～6cを、他の計算機2b上で実行させる。



【特許請求の範囲】

【請求項1】 複数の計算機が結合された環境で動作するプログラムの動作状況を監視するクラスタシステムであって、

前記環境で動作するプログラムの識別情報を取得する識別情報取得手段と、

当該識別情報取得手段によって取得された識別情報で示されるプログラムが正常か否かを監視する監視手段と、当該監視手段による監視によって前記プログラムの異常が検出された場合に、この異常の検出されたプログラムを実行していた計算機上でこの異常の検出されたプログラムを再起動させる再起動手段と、

当該再起動手段によって再起動されたプログラムの異常が検出された場合に、当該異常の検出されたプログラムを実行していた計算機上で動作していたプログラムを、他の計算機上で実行させるプログラム移転手段とを具備したことを特徴とするクラスタシステム。

【請求項2】 複数の計算機が結合された環境で動作しアプリケーション・インターフェースを持つアプリケーション・プログラムの動作状況を監視するクラスタシステムであって、

前記環境で動作するアプリケーション・プログラムの識別情報を取得する識別情報取得手段と、

当該識別情報取得手段によって取得された識別情報で示されるアプリケーション・プログラムのアプリケーション・インターフェイスを呼び出して正常に実行されているか否かを監視する監視手段と、

当該監視手段による監視によって前記アプリケーション・プログラムの異常が検出された場合に、この異常の検出されたアプリケーション・プログラムを実行していた計算機上でこの異常の検出されたアプリケーション・プログラムを再起動させる再起動手段と、

当該再起動手段によって再起動されたアプリケーション・プログラムの異常が検出された場合に、当該異常の検出されたアプリケーション・プログラムを実行していた計算機上で動作していたプログラムを、他の計算機上で実行させるプログラム移転手段とを具備したことを特徴とするクラスタシステム。

【請求項3】 コンピュータに、

複数の計算機が結合された環境で動作するプログラムの識別情報を取得する識別情報取得機能と、

当該識別情報取得機能によって取得された識別情報で示されるプログラムが正常か否かを監視する監視機能と、

当該監視機能による監視によって前記プログラムの異常が検出された場合に、この異常の検出されたプログラムを実行していた計算機上でこの異常の検出されたプログラムを再起動させる再起動機能と、

当該再起動機能によって再起動されたプログラムの異常が検出された場合に、当該異常の検出されたプログラム

を実行していた計算機上で動作していたプログラムを、他の計算機上で実行させるプログラム移転機能とを実現させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体。

【請求項4】 コンピュータに、
複数の計算機が結合された環境で動作しアプリケーション・インターフェースを持つアプリケーション・プログラムの識別情報を取得する識別情報取得機能と、
当該識別情報取得機能によって取得された識別情報で示されるアプリケーション・プログラムのアプリケーション・インターフェイスを呼び出して正常に実行されているか否かを監視する監視機能と、
当該監視機能による監視によって前記アプリケーション・プログラムの異常が検出された場合に、この異常の検出されたアプリケーション・プログラムを実行していた計算機上でこの異常の検出されたアプリケーション・プログラムを再起動させる再起動機能と、
当該再起動機能によって再起動されたアプリケーション・プログラムの異常が検出された場合に、当該異常の検出されたアプリケーション・プログラムを実行していた計算機上で動作していたプログラムを、他の計算機上で実行させるプログラム移転機能とを実現させるためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、複数の計算機が結合された環境における各計算機の動作を監視するクラスタシステム及びプログラムを記憶したコンピュータ読み取り可能な記憶媒体に関する。

【0002】

【従来の技術】クラスタシステムは、ネットワークを介して接続された計算機（コンピュータ）にクラスタ・ソフトウェアを搭載して構築される。

【0003】図5は、従来のクラスタシステムの概略を示すブロック図である。このクラスタシステム1では、2台の計算機（コンピュータ）2a、2bが通信回線3で接続されており、各計算機2a、2b上ではオペレーティング・システム（以下、「OS」という）4とクラスタ・ソフトウェア5とが実行されている。

【0004】クラスタシステム1を構成する各計算機2a、2bは、例えばデータベース管理プログラム、電子メール管理プログラム、ディレクトリ・サービス提供用プログラム、通信プログラムのようなアプリケーション・プログラム（以下、「アプリケーション」という）を実行可能である。図5では、計算機2a上でアプリケーション6a～6cが実行されている場合を例として示している。

【0005】このような各種アプリケーション6a～6cをクラスタシステム1に導入する際には、各アプリケ

ーション6a～6c専用の監視プログラム（監視モジュール）7a～7cをクラスタ・ソフトウェアに追加する必要がある。

【0006】このアプリケーション6a～6c専用の監視プログラム7a～7cは、それぞれのアプリケーション6a～6cが実行されている場合に、OS4のコマンドにより定期的に実行状態を調査する。

【0007】また、監視プログラム7a～7cは、このOS4のコマンドによる調査の結果、監視対象のアプリケーション6a～6cの動作異常や、監視対象のアプリケーション6a～6cを実行している計算機2aの異常を検出すると、この監視対象のアプリケーション6a～6cの再起動を行う。

【0008】さらに、監視プログラム7a～7cは、この再起動が失敗した場合には、オペレータからの指示にしたがって、監視対象のアプリケーション6a～6cを他の計算機2b上で起動させる処理を実行する。

【0009】すなわち、クラスタ・ソフトウェアは、オペレータに指定されたアプリケーション6a～6cを立ち上げ、定期的に互いの計算機2a、2bが正常か否かを監視し、また計算機2a、2b上で実行されているアプリケーション6a～6cが正常か否かを監視する。さらに、監視の結果、障害発生を検出すると、この障害に関係のあるアプリケーションの再起動を試み、再起動後さらに障害が発生した場合には、その旨をオペレータに通知する。その後、クラスタ・ソフトウェアは、オペレータからの指示があれば、障害の発生した計算機2a上のデータやプログラムを他の計算機2bに引き継がせる。

【0010】上記のような従来のクラスタシステム1の動作について説明する。従来のクラスタシステム1上で動作している各種アプリケーション6a～6cは、アプリケーション6a～6c自体の問題、OS4やハードウェアの問題、オペレータの操作誤りなどによって、正常に動作しなくなったり、計算機2aから消滅する場合がある。

【0011】このような場合に備えて、計算機2aのオペレータや監視プログラム7a～7cは、OS4が提供する監視用コマンドを実行したり、ログメッセージを参照することで、各種アプリケーション6a～6cあるいは計算機2aの動作状況を監視している。

【0012】この動作状況監視の結果、何らかの問題が検出された場合に、オペレータや監視プログラム7a～7cは、その問題の種類に対応して異常とされたアプリケーションを再起動させたり、計算機2aを再立ち上げを行う。あるいは、複数台の計算機2a、2bが疎結合されたクラスタシステム1においては、オペレータが指示を発して、障害の発生した計算機2aと異なる他の計算機2b上で障害の発生したアプリケーションが引き継がれる。

【0013】

【発明が解決しようとする課題】上記のような図5に例示される従来のクラスタシステム1においては、動作中のアプリケーション6a～6cのいずれかに異常が発生した場合に、オペレータや監視プログラム7a～7cは、発生した異常の状況に応じて、計算機2a上で異常の発生したアプリケーションの再起動を行う。そして、それでも異常が発生する場合には、他系の計算機2bにこの異常の発生するアプリケーションの引き継ぎを行う。

【0014】しかしながら、この異常の発生したアプリケーションの他系の計算機2bへの引き継ぎはオペレータの判断で行われるので、オペレータがアプリケーション6a～6cに関するエラーメッセージを見落したり、監視プログラム7a～7cが発生させるエラーメッセージを見落とすことがある。

【0015】この場合、他系の計算機2bへの引き継ぎが遅れ、当該クラスタシステム1の運用に支障をきたす場合がある。また、これにより、クラスタシステム1の信頼性、可用性が低下する場合がある。

【0016】また、従来のクラスタシステム1においては、導入するアプリケーション6a～6c用の監視プログラム7a～7cをオペレータがクラスタ・ソフトウェアに組み込まなければならないため、アプリケーション導入時にオペレータに手間がかかるという問題がある。

【0017】本発明は、以上のような実情に鑑みてなされたもので、クラスタシステムに容易に新規のプログラムを導入可能であり、何らかの異常が発生しても継続してプログラムを実行可能なクラスタシステム及びプログラムを記憶したコンピュータ読み取り可能な記憶媒体を提供することを目的とする。

【0018】

【課題を解決するための手段】本発明の骨子は、クラスタシステムを構成する計算機上で動作するプログラムをプログラムの種別に依存することなくまとめて監視する手段を備えた点にある。また、異常検出後に再起動したプログラムの異常をさらに検出した場合に、この異常を検出したプログラムを実行していた計算機上で動作していたプログラムのうち移動可能なプログラム全てを他の計算機上で起動させる点にある。

【0019】以下、本発明を実現するにあたって講じた具体的手段について説明する。

【0020】第1の発明は、複数の計算機が結合された環境で動作するプログラムの動作状況を監視するクラスタシステムに関する発明である。

【0021】この第1の発明のクラスタシステムは、この環境で動作するプログラムの識別情報を取得する識別情報取得手段と、当該識別情報取得手段によって取得された識別情報で示されるプログラムが正常か否かを監視

する監視手段と、当該監視手段による監視によってプログラムの異常が検出された場合に、この異常の検出されたプログラムを実行していた計算機上でこの異常の検出されたプログラムを再起動させる再起動手段と、当該再起動手段によって再起動されたプログラムの異常が検出された場合に、当該異常の検出されたプログラムを実行していた計算機上で動作していたプログラムを他の計算機上で実行させるプログラム移転手段とを具備する。

【0022】すなわち、この第1の発明のクラスタシステムにおいては、動作状態にあるプログラムの識別情報が収集され、動作中のプログラムが自動的にまとめて監視される。

【0023】したがって、新規のプログラムを導入する場合であっても、この新規のプログラム専用の監視プログラムを導入する必要がない。

【0024】ゆえに、専用の監視プログラムを特別に用意しなくても、新規のプログラムを容易に導入できる。

【0025】また、この第1の発明のクラスタシステムにおいては、異常の生じたプログラムを再起動してもさらに異常が発生する場合には、この異常の発生する計算機上で実行されていたプログラムが自動的に他の計算機上で起動される。

【0026】したがって、オペレータが異常に気づかなければ、プログラムの引き継ぎが遅れることを防止でき、高い信頼性、可用性を確保することができる。

【0027】第2の発明は、複数の計算機が結合された環境で動作しアプリケーション・インターフェースを持つアプリケーション・プログラムの動作状況を監視するクラスタシステムに関する発明である。

【0028】この第2の発明のクラスタシステムは、この環境で動作するアプリケーション・プログラムの識別情報を取得する識別情報取得手段と、当該識別情報取得手段によって取得された識別情報で示されるアプリケーション・プログラムのアプリケーション・インターフェイスを呼び出して正常に実行されているか否かを監視する監視手段と、当該監視手段による監視によってアプリケーション・プログラムの異常が検出された場合に、この異常の検出されたアプリケーション・プログラムを実行していた計算機上でこの異常の検出されたアプリケーション・プログラムを再起動させる再起動手段と、当該再起動手段によって再起動されたアプリケーション・プログラムの異常が検出された場合に、当該異常の検出されたアプリケーション・プログラムを実行していた計算機上で動作していたプログラムを他の計算機上で実行させるプログラム移転手段とを具備する。

【0029】この第2の発明においては、動作中のアプリケーションの識別情報が管理され、この動作中のアプリケーションの含むアプリケーション・インターフェイスに対して適宜呼び出しが行われる。この呼び出しの結果、例えばこの呼び出しに関する応答がない場合や、あ

るいは応答としてエラーを受けた場合には、このアプリケーションが異常とされる。

【0030】したがって、各アプリケーション専用の監視プログラムが必要ないため、上記第1の発明と同様の作用効果を得ることができる。

【0031】また、この第2の発明のクラスタシステムにおいては、異常の生じたアプリケーション・プログラムを再起動してもさらに異常が発生する場合には、他の計算機上で起動される。

【0032】したがって、上記第1の発明の場合と同様に、オペレータが異常に気づかなければ、プログラムの引き継ぎが遅れることを防止でき、高い信頼性、可用性を確保することができる。

【0033】第3の発明は、コンピュータに、複数の計算機が結合された環境で動作するプログラムの識別情報を取得させる識別情報取得機能と、当該識別情報取得機能によって取得された識別情報で示されるプログラムが正常か否かを監視させる監視機能と、当該監視機能による監視によってプログラムの異常が検出された場合に、この異常の検出されたプログラムを実行していた計算機上でこの異常の検出されたプログラムを再起動させる再起動機能と、当該再起動機能によって再起動されたプログラムの異常が検出された場合に、当該異常の検出されたプログラムを実行していた計算機上で動作していたプログラムを他の計算機上で実行させるプログラム移転機能とを実現させるプログラムを記憶したコンピュータ読み取り可能な記憶媒体である。

【0034】また、第4の発明は、コンピュータに、複数の計算機が結合された環境で動作しアプリケーション・インターフェースを持つアプリケーション・プログラムの識別情報を取得させる識別情報取得機能と、当該識別情報取得機能によって取得された識別情報で示されるアプリケーション・プログラムのアプリケーション・インターフェイスを呼び出して正常に実行されているか否かを監視させる監視機能と、当該監視機能による監視によってアプリケーション・プログラムの異常が検出された場合に、この異常の検出されたアプリケーション・プログラムを実行していた計算機上でこの異常の検出されたアプリケーション・プログラムを再起動させる再起動機能と、当該再起動機能によって再起動されたアプリケーション・プログラムの異常が検出された場合に、当該異常の検出されたアプリケーション・プログラムを実行していた計算機上で動作していたプログラムを他の計算機上で実行させるプログラム移転機能とを実現させるプログラムを記憶したコンピュータ読み取り可能な記憶媒体である。

【0035】第3及び第4の発明は、それぞれ第1及び第2の発明で説明したクラスタシステムの機能をコンピュータにより実現するためのプログラムを記憶したコンピュータ読み取り可能な記憶媒体である。

【0036】このようなプログラムを記憶した記憶媒体を用いることによって、上述した機能を有していない計算機や計算機システムに対しても、簡単に上述した機能を付加することができる。

【0037】

【発明の実施の形態】以下、図面を参照しながら本発明の実施の形態について説明する。

【0038】(第1の実施の形態) 本実施の形態においては、動作中のプログラムをまとめて監視し、何らかの異常が発生したらそのプログラムを再起動し、それでも異常が発生するようであればプログラムを他系の計算機に自動的に引き継がせるクラスタシステムについて説明する。

【0039】図1は、本実施の形態に係るクラスタシステムの概略を例示するブロック図であり、図5と同一の部分については同一の符号を付してその説明を省略するかあるいは簡単に説明し、ここでは異なる部分についてのみ詳しく説明する。

【0040】クラスタシステム8を構成する一方の計算機2aは、通常時にアプリケーションを動作させるコンピュータである。これに対し、他方の計算機2bは、待機系として動作するコンピュータであり、一方の計算機2aに異常が発生した場合にその業務を引き継ぐ。通信回線3は、計算機2a、2b間を送受信可能に接続している。

【0041】なお、計算機2a、2bにはそれぞれ記憶領域9が設けられているが、図1では計算機2b側の記憶領域の記載は省略している。

【0042】OS4は、各計算機2a、2bを制御するソフトウェアであり、アプリケーション6a～6cは、各種の業務を処理するためのプログラムである。ここでは、この各アプリケーション6a～6cは計算機に常駐した形態で利用されるプログラムとする。また、このアプリケーション6a～6cは、動作する計算機を移動可能なプログラムとする。

【0043】クラスタ・ソフトウェア10は、主に、クラスタ・ソフトウェア本体11と監視プログラム12から構成されており、計算機2a、2b双方に備えられている。以下においては、計算機2a側のクラスタソフトウェア10を例として説明するが、計算機2b側のクラスタソフトウェア10も同様の機能を持ち、同様の動作を行う。

【0044】クラスタ・ソフトウェア本体11は、先において説明した従来のクラスタ・ソフトウェア5と同様の機能を保持しており、その中には識別情報取得機能11a、プログラム移転機能11b、計算機停止機能11cが含まれている。

【0045】識別情報取得機能11aは、この計算機2a上で動作すべきアプリケーションのプロセスIDを記憶領域9に保持する。これにより、計算機2a上で動作

しているはずのアプリケーションが特定される。

【0046】プログラム移転機能11bは、計算機2a上で動作していたアプリケーションを他の計算機2b上で動作させるための処理を行い、また計算機2b上で動作していたアプリケーションを計算機2a上で動作させるための処理を行う。計算機停止機能11cは、計算機2aを停止するための処理を行う。

【0047】監視プログラム12は、監視機能12a、再起動機能12b、プログラム移転指示機能12cとを含む。監視機能12aは、記憶領域9を参照し、この記憶領域9にプロセスIDが登録されているアプリケーション(計算機2a上に存在すべきアプリケーション)が計算機2a上に存在するか否かを、OS4の提供するコマンドを呼び出すことで確認する。

【0048】再起動機能12bは、監視機能12aによって計算機2a上で動作していたアプリケーションの異常終了が検出された場合に、この異常終了したアプリケーションを当該計算機2a上で再起動させる。

【0049】プログラム移転指示機能12cは、再起動機能12bによって再起動されたアプリケーションの異常終了が再び監視機能12aによって検出された場合に、計算機2a、2bのクラスタ・ソフトウェア本体11の持つプログラム移転機能11bに対し、この計算機2a上で動作するアプリケーション6a～6cの移転指示(計算機b上でのアプリケーション6a～6cの起動・開始指示)を通知する。

【0050】同様に、プログラム移転指示機能12は、再起動機能12bによって再起動されたアプリケーションの異常終了が再び監視機能12aによって検出された場合に、計算機2aのクラスタソフトウェア本体11の持つ計算機停止機能11cに計算機2aの停止指示を通知する。

【0051】上記のような構成を持つクラスタシステム8の動作について以下に説明する。図2は、本実施の形態に係るクラスタシステム8の動作を示すフロー図であり、特に監視プログラム12によるアプリケーション6a～6cの存在確認と、アプリケーション6a～6cの消滅検出時の処理手順を示している。

【0052】クラスタシステム8においては、まず、監視プログラム12の監視機能12aによってプロセスIDが登録されている記憶領域9が参照され、監視すべきアプリケーション6a～6cのプロセスIDが取り出される(s1)。

【0053】次に、監視プログラム12の監視機能12aによってOS4の提供するコマンドが呼び出され、計算機2a上で動作しているアプリケーションのプロセスIDが取り出される(s2)。これにより、計算機2a上で動作しているアプリケーションのプロセスIDの一覧情報が得られる。

【0054】次に、記憶領域9から得られたプロセスI

D（処理s1で得られたプロセスID）と、OS4のコマンド呼び出しにより得られたプロセスID（処理s2で得られたプロセスID）とが、監視プログラム12の監視機能12aにより比較される。そして、計算機2aで動作しているべき監視対象のアプリケーション6a～6cが消滅しているか否かが判定される（s3）。

【0055】この比較の結果、監視すべきアプリケーション6a～6cが消滅していない場合には、上記の監視が繰り返される。

【0056】監視すべきアプリケーション6a～6cのいずれかが消滅している場合には、再起動機能12bにより消滅したアプリケーションが先において一度再起動されたものか否かが判定される（s4）。

【0057】判定の結果、未だ再起動されていない場合には、消滅したアプリケーションが監視プログラム12の再起動機能12bによって消滅前と同一の計算機2a上で再起動される（s5）。

【0058】一方、消滅したアプリケーションが再起動済みであった場合には、監視プログラム12のプログラム移転指示機能12cにより、計算機2aのクラスタ・ソフトウェア10の持つ計算機停止機能11cに計算機2aの停止が指示される。また、再起動しても異常が発生するのはアプリケーションよりむしろ他の異常（OS、計算機の異常）である可能性が高いとして、計算機2a上で動作していたアプリケーション6a～6cの計算機2b上での起動が、計算機2a、2bのクラスタ・ソフトウェア10の持つプログラム移転機能11bに指示される（s6）。

【0059】この指示により、計算機2aのクラスタ・ソフトウェア10の持つ計算機停止機能11cが計算機2aを停止させる。また、計算機2a、2bのクラスタ・ソフトウェア10の持つプログラム移転機能11bによりアプリケーション6a～6cの動作が計算機2bに引き継がれる。

【0060】以上説明したように、本実施の形態に係るクラスタシステム8においては、アプリケーション6a～6cをまとめて監視する監視プログラム12をクラスタ・ソフトウェア10に加えている。

【0061】したがって、新規のアプリケーションを導入するたびに、このアプリケーション専用の監視プログラムをオペレータが加える必要がなく、オペレーターの作業を軽減させることができる。

【0062】また、本実施の形態に係るクラスタシステム8においては、アプリケーションに対する再起動後、さらに再起動が失敗した場合には、この再起動が失敗した計算機上で動作していた移動可能なアプリケーションの動作を全て待機系の他の計算機に引き継がせる。

【0063】したがって、アプリケーションの停止状態が持続することを防止し、システムの信頼性、可用性が向上される。

【0064】なお、本実施の形態においては、監視対象のプログラムが常駐形式で動作するアプリケーション6a～6cの場合を例として説明しているが、これに限定されるものではない。例えば、アプリケーションではないデーモンや、常駐形式ではないアプリケーションを監視対象とする場合でも、同様の手法を適用することで同様の効果を得ることができる。また、監視対象のプログラムの数にも制限されることなく、いくつでもよい。

【0065】また、本実施の形態においては、2台の計算機2a、2bによりクラスタシステム8が構成される場合を例として説明しているが、これに限定されるものではなく、3台以上の計算機によりクラスタシステムが構成される場合にも同様に適用可能である。

【0066】また、本実施の形態に係るクラスタシステム8は、同様の作用・機能を実現可能であれば各構成要素の配置を変更させてもよく、また各構成要素を自由に組み合わせてもよい。例えば、識別情報取得機能11aは、クラスタ・ソフトウェア本体11に備えるのではなく、監視プログラム12に備えてもよい。

【0067】（第2の実施の形態）本実施の形態においては、アプリケーション・インターフェイス（以下、「API」という）を持つアプリケーションを監視対象とし、このアプリケーションを他系の計算機に自動的に引き継がせるクラスタシステムについて説明する。

【0068】図3は、本実施の形態に係るクラスタシステムの概略を例示するブロック図であり、図1、5と同一の部分については同一の符号を付してその説明を省略するかあるいは簡単に説明し、ここでは異なる部分についてのみ詳しく説明する。

【0069】本実施の形態に係るクラスタシステム13の基礎的な構成は、先で述べたクラスタシステム8と同様であるが、アプリケーションのAPIを利用して監視を行う点が異なる。

【0070】すなわち、アプリケーション14a～14cは、それぞれに対するインターフェイスを扱うAPI15a～15cを備えている。API15a～15cは、外部のプログラムにアプリケーション14a～14cの機能を利用させるために動作する。例えば、アプリケーション14aがデータベース管理プログラムである場合には、API15aはSQL命令を受け付け、処理後の結果を返す。

【0071】クラスタ・ソフトウェア16の監視プログラム17は、監視対象のアプリケーション14a～14cの各API15a～15cを呼び出す。その結果、何らかのエラーがアプリケーション14a～14cの持つAPI15a～15cのいずれかから返却されたり、あるいはAPI15a～15bのいずれかから制御がリターンしなくなった場合、監視プログラム17は、異常を検出したAPIを持つアプリケーションを停止させ、再起動させる。

【0072】また、この監視プログラム17は、再起動後のAPI呼び出しでまだ異常を検出する場合には、計算機2aのクラスタ・ソフトウェア本体11に対して、計算機2aの停止を指示し、さらに計算機2a、2bのクラスタ・ソフトウェア本体11に対して、計算機2b上でのアプリケーション14a～14cの起動・開始を指示する。

【0073】上記のような構成を持つクラスタシステム13の動作について以下に説明する。図4は、本実施の形態に係るクラスタシステム13の動作を示すフロー図であり、特に監視プログラム17によるアプリケーション14a～14cの持つAPI15a～15cの呼び出しと、API15a～15cの呼び出しにおいて異常を検出した際の処理手順を示している。

【0074】このクラスタシステム13においては、まず、監視プログラム17の監視機能によって記憶領域9が参照され、この監視プログラム17によって監視すべきアプリケーション14a～14bの持つAPI15a～15cが呼び出される(t1)。

【0075】ここで、このAPI15a～15bの呼び出しに対して、API15a～15cのいずれかからエラー返却を受けたか、あるいはリターンを返却しない(制御戻さない)かの判定が、監視プログラム17の監視機能によって行われる(t2)。

【0076】正常にリターンを受けた場合には、上記の処理が繰り返されるが、エラー返却を受けたりリターンを返却しない場合には、そのAPIを持つアプリケーションが再起動済みか否かが監視プログラム17の再起動機能により判定される(t3)。

【0077】判定の結果、未だ再起動されていない場合には、この正常なリターンを返さないAPIを持つアプリケーションが、監視プログラム17の再起動機能によって同一の計算機2a上で再起動される(t4)。

【0078】一方、既に再起動済みの場合には、監視プログラム17のプログラム移転指示機能により、計算機2aの停止が計算機2aのクラスタ・ソフトウェア本体11に指示される。また、計算機2a上で動作していたアプリケーション14a～14cの計算機2b上での起動が、計算機2a、2bのクラスタソフトウェア本体11に指示される(t5)。

【0079】この指示により、計算機2aのクラスタ・ソフトウェア16の持つ計算機停止機能が計算機2aを停止させる。また、計算機2a、2bのクラスタ・ソフトウェア16の持つプログラム移転機能によりアプリケーション14a～14cの動作が計算機2bに引き継がれる。

【0080】以上説明したように、本実施の形態に係るクラスタシステム13においては、監視プログラム17がアプリケーション14a～14cのAPI呼び出しにより、まとめてアプリケーション14a～14cの監視

を行う。

【0081】また、アプリケーション14a～14cのAPI呼び出しに対する異常動作、ストールを検出した場合に当該異常の発生したアプリケーションの再起動が実行され、さらに異常がある場合に待機系の計算機2bへアプリケーション14a～14cの業務が自動的に引き継がれる。

【0082】これにより、先で述べた第1の実施の形態と同様に、オペレータの作業を軽減させ、さらにシステムの信頼性、可用性が向上される。

【0083】なお、本実施の形態においては、2台の計算機2a、2bによりクラスタシステム13が構成される場合を例として説明しているが、これに限定されるものではなく、3台以上の計算機によりクラスタシステムが構成される場合にも同様に適用可能である。同様に、クラスタシステム上で動作するアプリケーションの数にも、特に制限はなくいくつであってもよい。

【0084】また、本実施の形態に係るクラスタシステム13は、同様の作用・機能を実現可能であれば各構成要素の配置を変更させてもよく、また各構成要素を自由に組み合わせてもよい。

【0085】また、上記第1及び第2の実施の形態に係るクラスタシステム8、13におけるクラスタ・ソフトウェア10、16は、コンピュータに実行させることのできるプログラムとして、例えば磁気ディスク(フロッピー(登録商標)ディスク、ハードディスク等)、光ディスク(CD-ROM、DVD等)、半導体メモリなどの記憶媒体に書き込んで適用したり、通信媒体により伝送して計算機あるいは計算機システムに適用することも可能である。上記各機能を実現するコンピュータは、記憶媒体に記憶されたプログラムを読み込み、プログラムによって動作が制御されることにより、上述した処理を実行する。

【0086】

【発明の効果】以上詳記したように本発明においては、プログラムの種別に関係なく、計算機上で動作するプログラムが正常か否かを監視する監視手段と、異常の発生したプログラムの再起動に失敗した場合に、その計算機上で動作する移動可能なプログラムを他の計算機に自動的に移転するプログラム移転手段とを備えている。

【0087】このように、監視手段がプログラムの種別に関係なく各プログラムの動作をまとめて監視することで、各プログラム専用の監視プログラムを導入する必要がない。

【0088】したがって、新規のプログラム導入時のオペレータの労力を低減させることができる。

【0089】また、自動的に再起動できなかったプログラムを他の計算機に移転することで、プログラムの移転が遅れることを防止することができ、これによりシステムの信頼性、可用性を向上させることができる。

【0090】また、再起動しても異常が発生するのは、プログラムの異常よりも計算機やOSの異常である可能性が高いため、この計算機上の移動可能なプログラムを全て移転することでさらに信頼性、可用性を向上させることができる。

【図面の簡単な説明】

【図1】本発明の第1の実施の形態に係るクラスタシステムの概略を例示するブロック図。

【図2】同実施の形態に係るクラスタシステムの動作を示すフロー図。

【図3】本発明の第2の実施の形態に係るクラスタシステムの概略を例示するブロック図。

【図4】同実施の形態に係るクラスタシステムの動作を示すフロー図。

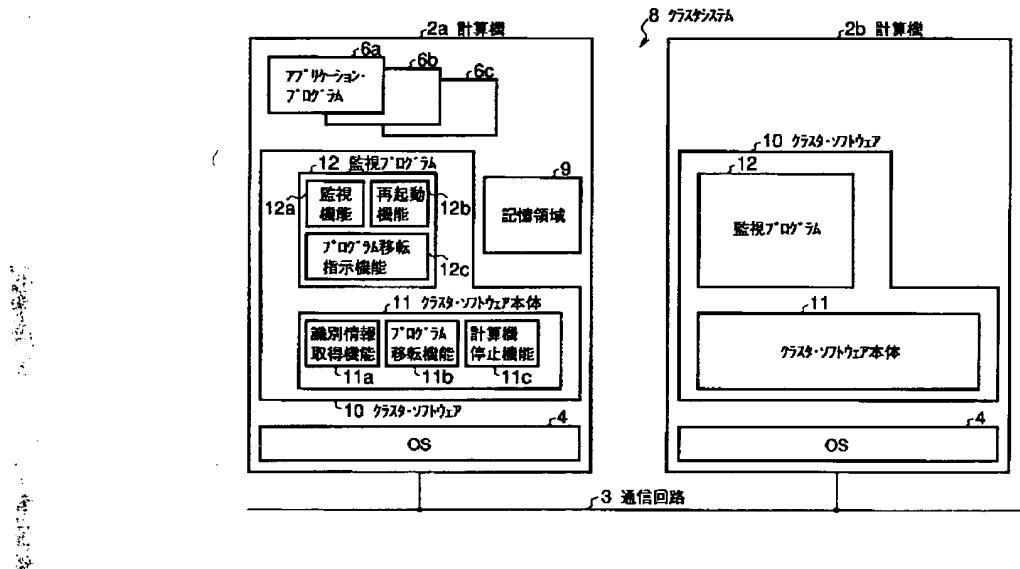
【図5】従来のクラスタシステムの概略を例示するブロック図。

【符号の説明】

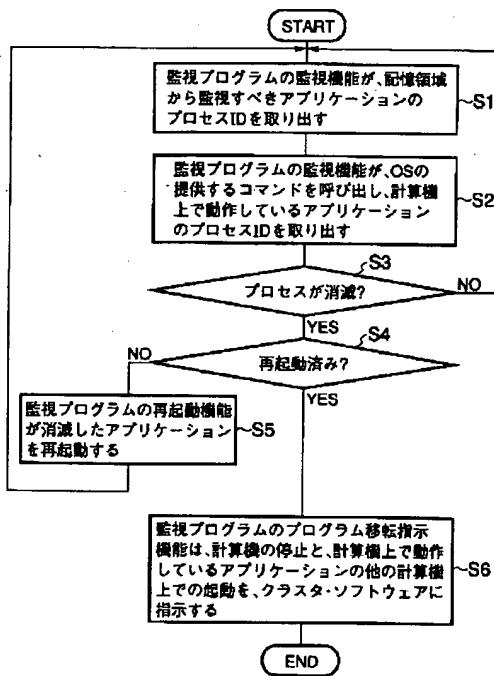
1、8、13…クラスタシステム

- 2a、2b…計算機
- 3…通信回線
- 4…オペレーティング・システム
- 5、10、16…クラスタ・ソフトウェア
- 6a～6c、14a～14c…アプリケーション・プログラム
- 15a～15c…アプリケーション・インターフェイス
- 7a～7c…アプリケーション専用監視プログラム
- 9…記憶領域
- 11…クラスタ・ソフトウェア本体
- 11a…識別情報取得機能
- 11b…プログラム移転機能
- 11c…計算機停止機能
- 12、17…監視プログラム
- 12a…監視プログラム
- 12b…再起動機能
- 12c…プログラム移転指示機能
- 12d…記憶領域
- 11…クラスタ・ソフトウェア本体
- 11a…識別情報取得機能
- 11b…プログラム移転機能
- 11c…計算機停止機能
- 10…クラスタ・ソフトウェア
- 4…OS
- 3…通信回路

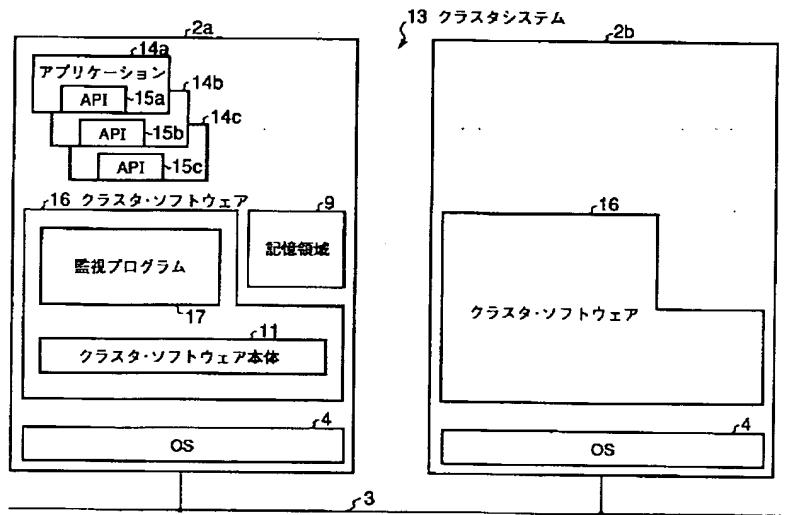
【図1】



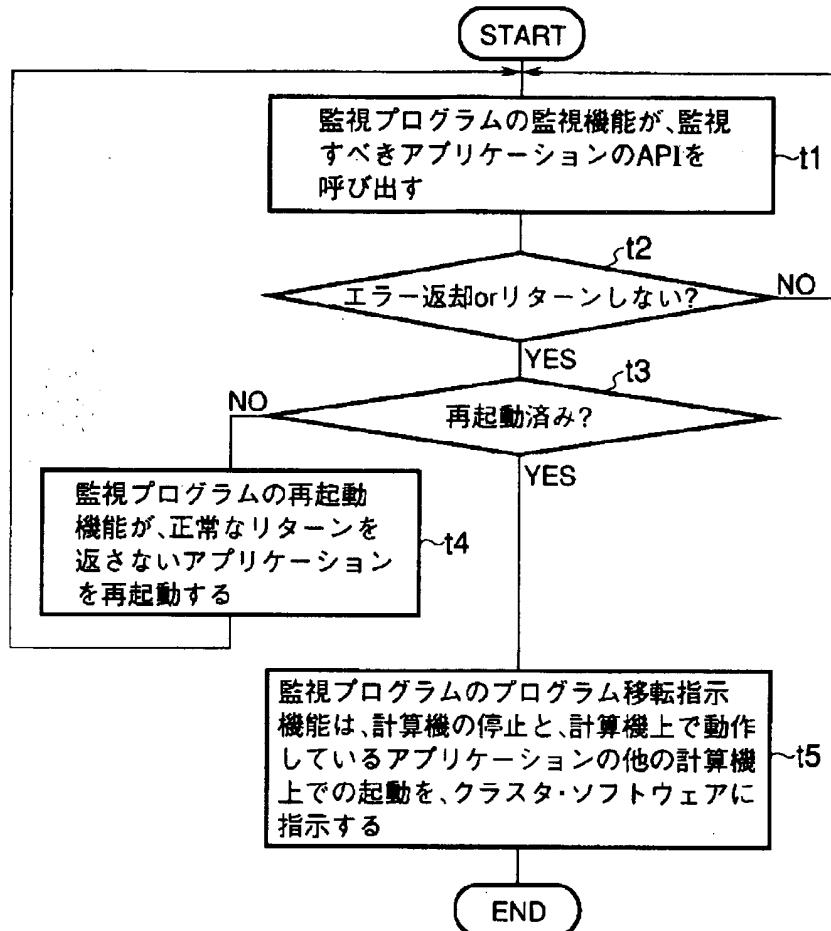
【図2】



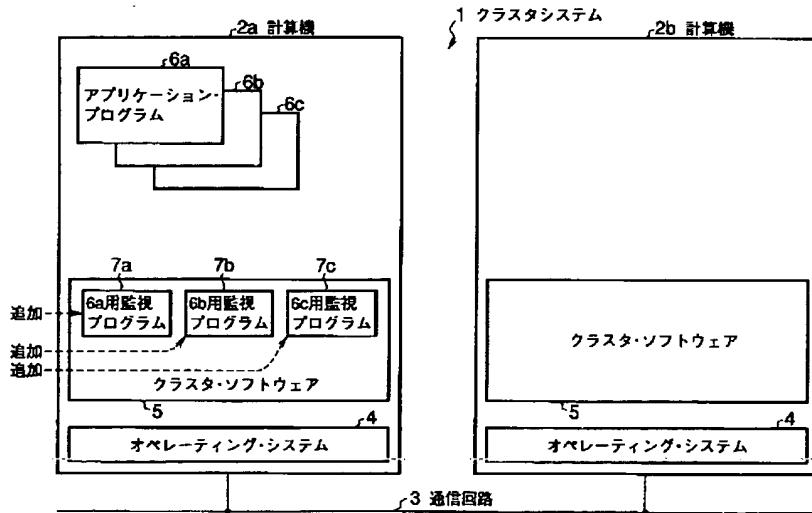
【図3】



【図4】



【図5】



(11) 月2001-22709 (P2001-227JL

フロントページの続き

(51) Int.Cl. ⁷	識別記号	F I	マークド(参考)
G 06 F 15/177	678	G 06 F 15/177	678B
			678A

F ターム(参考) 5B034 BB02 CC01 DD02
5B042 GA11 JJ15 KK05
5B045 GG01 JJ02 JJ44 JJ45
5B098 AA10 GA02 GC01 JJ02 JJ08

